JP2002084302

Publication Title:

METHOD AND APPARATUS FOR COMMUNICATION BY NETWORK

Abstract:

Abstract of JP2002084302

PROBLEM TO BE SOLVED: To provide a method and an apparatus for communication by a virtual network, where independent protocol processing for each RS is realized, and a plurality of independent virtual networks can be realized. SOLUTION: An OS or a service layer individually recognizes a plurality of virtual data links, constituted on a physical data link, and the virtual data links are relayed independently.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－84302
（P2002－84302A）
(43)公開日　平成14年３月22日(2002.3.22)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード*（参考）
Ｈ０４Ｌ　12/46		Ｃ０６Ｆ　9/46	３４０Ａ	５Ｂ０８９
12/28		13/00	３５３Ｃ	５Ｂ０９８
Ｇ０６Ｆ　9/46	３４０	Ｈ０４Ｌ　11/00	３１０Ｃ	５Ｋ０３０
13/00	３５３	11/20	１０２Ｄ	５Ｋ０３３
Ｈ０４Ｌ　12/56		13/00	３０３Ｂ	５Ｋ０３４

審査請求　未請求　請求項の数11　ＯＬ　（全　９　頁）　最終頁に続く

(21)出願番号　特願2000－270778(P2000－270778)

(22)出願日　平成12年９月６日(2000.9.6)

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区大手町二丁目３番１号

(72)発明者　廣津　登志夫
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

(72)発明者　福田　健介
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

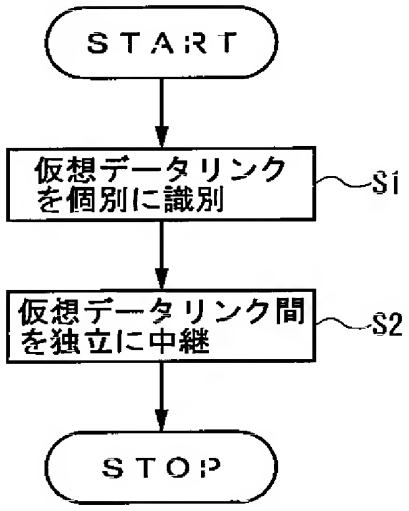
(74)代理人　100070150
弁理士　伊東　忠彦

最終頁に続く

(54)【発明の名称】　ネットワークによる通信方法及び装置

(57)【要約】
【課題】　ＲＳ毎の独立したプロトコル処理を実現し、独立した複数の仮想ネットワークを実現することが可能な仮想ネットワークによる通信方法及び装置を提供する。
【解決手段】　本発明は、ＯＳやサービスレイヤが物理データリンク上に構成される複数の仮想データリンクを個別に認識し、仮想データリンク間を独立に中継する。

本発明の原理を説明するための図



【特許請求の範囲】
【請求項１】　インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、
ＯＳ（オペレーティングシステム）やサービスレイヤが物理データリンク上に構成される複数の仮想データリンクを個別に認識し、
前記仮想データリンク間を独立に中継することを特徴とするネットワークによる通信方法。
【請求項２】　前記仮想データリンク間を独立に中継する際に、
ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する請求項１記載のネットワークによる通信方法。
【請求項３】　前記仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける請求項１記載のネットワークによる通信方法。
【請求項４】　インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、
複数の仮想データリンクや物理データリンを関連付けてグループ化したＯＳ内の処理モジュールをリソーススペース（以下、ＲＳ）とし、
前記ＲＳ毎に独立したプロトコル処理を行うことを特徴とするネットワークによる通信方法。
【請求項５】　通信の発信元や送信元のＩＰアドレスやポート番号を含む上位層の情報に応じて、通信を前記ＲＳに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける請求項４記載のネットワークによる通信方法。
【請求項６】　インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、
ＯＳ（オペレーティングシステム）やサービスレイヤで実現する物理データリンク上に構成される複数の仮想データリンクを個別に認識する認識手段と、前記仮想データリンク間を独立に中継する中継手段とを有することを特徴とす
るネットワークによる通信装置。
【請求項７】　前記中継手段は、
ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する手段を含む請求項６記載のネットワークによる通信装置。
【請求項８】　前記仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける手段を有する請求項６記載のネットワークによる通信装置。
【請求項９】　インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、
複数の仮想データリンクや物理データリンを関連付けてグループ化したＯＳ内の処理モジュールであるリソーススペース（以下、ＲＳ）を有し、
前記ＲＳ毎に独立したプロトコル処理を行うことを特徴とするネットワークによる通信装置。
【請求項１０】　通信の発信元や送信元のＩＰアドレスやポート番号を含む上位層の情報に応じて、通信を前記ＲＳに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける手段を有する請求項９記載のネットワークによる通信装置。
【請求項１１】　前記ＲＳは、
前記ＲＳ毎のポインタの集合であるポインタ表と、
アプリケーションのプロセスに付与されている前記ＲＳ毎に一意に付与されるＲＳ識別子（ＲＳＩＤ）に基づいて、前記ポインタ表を検索して呼び出されるシステムコール処理ルーチンと、
前記システムコール処理ルーチンの属するＲＳに基づいて特定され、該ＲＳから特定される仮想インタフェースを介してパケットを処理し、処理したパケットをネットワークに出力するプロトコル処理ルーチンとを有する請求項９記載のネットワークによる通信装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、ネットワークによる通信方法及び装置に係り、特に、単独の計算機、あるいは、ネットワークで接続された複数台からなる計算機、あるいは、インターネットに接続された計算機で構成されたシステムにおけるネットワークによる通信方法及び装置に関する。
【０００２】
【従来の技術】データリンク層においては、Ｅｔｈｅｒｎｅｔ（登録商標）（イーサネット（登録商標））やＡＴＭなどのように、一つの物理媒体の上に仮想的なデータリンクを複数構成する技術が実現されている。
【０００３】Ethernetでは、IEEE802.1Q規格により、伝送するEthernetのフレームに仮想データリンク（仮想ＬＡＮ：ＶＬＡＮ）の識別子となるタグという数字を保持させ、一つの物理媒体上に仮想的に複数のEthernetのネットワークを構成できる。Ehternetフレームを中継するEthernetスイッチでは、スイッチの各ポートに割り当てられているＶＬＡＮｉｄ（ＶＬＡＮ識別子）に応じて、送られてきたEthernetフレームを転送するかどうかを決定する。これにより、スイッチに接続された機器は所属するＶＬＡＮ以外のフレームを受信することはない。
【０００４】ＡＴＭでは、「Virtual Path(VP)」や「Virtual Connection(VC)」という概念で、機器の対の通信接続を、一つの物理媒体上に複数実現することができる。ネットワーク層やトランスポート層では、ＩＰ(Int

ernet Protocol）のＩＰアドレスや、ＴＣＰ(Transport Control Protocol)や、ＵＤＰ(User Datagram Protocol)のポート番号をチェックして、ネットワークの一部分に帰着する通信の中継を制限したり、ネットワークの一部分に帰着する通信以外すべての中継を制限したり、また、一部の通信の帯域を制御して、通信品質を変更することができる。これは、ルータのソフトウェアや、ホストＯＳ上の「ipfirewall」、「dummynet」のようなシステムで実現されている。アプリケーション層でも、ＩＰのＩＰアドレスやＴＣＰやＵＤＰのポート番号によって、ネットワークの一部分に帰着する通信を制限したりすることができる。これは、「ＴＣＰwrapper 」のようなソフトウェアで実現されている。x-kernel(N.C Hutchinscm and L.L Peterson."The x-kernel: An Architecture for Implementing Network Protocols", IEEE Transaction on Software Engineering, 17(1): 64-76, 1991)や、Scout(D Mosberger and L.L. Peterson. "Making Paths Explicit in the Scout Operating System", In proceeding of OSDI '96, 1996)といったＯＳでは、ＯＳ内の処理のモジュール化は行われている。また、Eclipse (J.bruno, E. Gabber, B. Ozden, and A. Silberschatz, "TheEclipse Operating System: Providing Quality of Serice via Reservation Domains", In Proceedings of the USENIX 1998 Annual Technical Conference,New Orleans, Louisiana, June 1998)においては、“Reservation Space ”（ＲＳ）という概念で、ＯＳ内部の計算資源の実時間性の確保が実現されている。

【０００５】

【発明が解決しようとする課題】しかしながら、仮想データリンクは、ルータやホストのプロトコル処理ソフトウェアでは、通常の物理データリンクと同様に扱われるので、従来のプロトコル処理ソフトウェアでは、ネットワーク層の処理ソフトウェアから、区別無く全ての仮想データリンク及び物理データリンクにアクセスすることができる。従って、仮想データリンク毎に異なる特性（通信品質やセキュリティレベル）を用意して、上位層のプロトコルやアプリケーションからそれらのデータリンクを区別して利用することはできない。

【０００６】また、上位層での通信制限は、ＩＰアドレスやポート番号により、仮想的にネットワークを構成していると考えることができ、仮想ネットワークに帰着する通信を区別して使用する。しかし、これらの技術では、通信の起点・終点の情報に応じて通信の制限をするだけで、それらに対してルータやホスト上の計算資源が独立して割り付けられているわけではない。

【０００７】「x-kernel」、「Scout 」、「Eclipse 」といったＯＳが実現しているのは、ＯＳの処理のモジュール化やそのグループ化であるが、これらは仮想ネットワークと連携しておらず、ネットワークは共有している。また、処理モジュールが概念的に仮想ネットワーク毎に分離しているわけではない。

【０００８】本発明は、上記の点に鑑みなされたもので、ホスト、ルータやスイッチなどのネットワーク機器において、複数の仮想データリンクや物理データリンクを関連付けてグループ化し、そのグループ内の仮想データリンク及び物理データリンクに使用する「処理プロトコル」、「計算資源」、「アプリケーション」を割り付けてグループ化する。この「データリンク」と「処理プロトコル」、「計算資源」、「アプリケーション」のグループを“Resource Space”（リソーススペース：ＲＳ）と呼ぶ。このＲＳ毎の独立したプロトコル処理を実現し、独立した複数の仮想ネットワークを実現することが可能なネットワークによる通信方法及び装置を提供することを目的とする。

【０００９】また、本発明の目的は、通信の発信元や送信先のＩＰアドレスやポート番号といった上位層の情報に応じて通信をＲＳに割り付けることにより、上位層の情報に応じて通信を適当な仮想ネットワークに割り付けることが可能なネットワークによる通信方法及び装置を提供することである。

【００１０】

【課題を解決するための手段】図１は、本発明の原理を説明するための図である。

【００１１】本発明（請求項１）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、ＯＳやサービスレイヤで実現する物理データリンク上に構成される複数の仮想データリンクを個別に認識し（ステップ１）、仮想データリンク間を独立に中継する（ステップ２）。

【００１２】本発明（請求項２）は、仮想データリンク間を独立に中継する際に、ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する。

【００１３】本発明（請求項３）は、仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける。

【００１４】本発明（請求項４）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、複数の仮想データリンクや物理データリンを関連付けてグループ化したＯＳ内の処理モジュールをＲＳとし、ＲＳ毎に独立したプロトコル処理を行う。

【００１５】本発明（請求項５）は、通信の発信元や送信元のＩＰアドレスやポート番号を含む上位層の情報に応じて、通信をＲＳに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける。

【００１６】本発明（請求項６）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、ＯＳやサービスレイヤで

実現する物理データリンク上に構成される複数の仮想データリンクを個別に認識する認識手段４００と、仮想データリンク間を独立に中継する中継手段３００とを有する。

【００１７】本発明（請求項７）は、中継手段３００において、ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する手段を含む。

【００１８】本発明（請求項８）は、仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける手段を有する。

【００１９】本発明（請求項９）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、複数の仮想データリンクや物理データリンクを関連付けてグループ化したＯＳ内の処理モジュールであるＲＳを有し、ＲＳ毎に独立したプロトコル処理を行う。

【００２０】本発明（請求項１０）は、通信の発信元や送信元のＩＰアドレスやポート番号を含む上位層の情報に応じて、通信をＲＳに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける手段を有する。

【００２１】本発明（請求項１１）は、ＲＳにおいて、ＲＳ毎のポインタの集合であるポインタ表と、アプリケーションのプロセスに付与されているＲＳ毎に一意に付与されるＲＳ識別子（ＲＳＩＤ）に基づいて、ポインタ表を検索して呼び出されるシステムコール処理ルーチンと、システムコール処理ルーチンの属するＲＳに基づいて特定され、該ＲＳから特定される仮想インタフェースを介してパケットを処理し、処理したパケットをネットワークに出力するプロトコル処理ルーチンとを有する。

【００２２】上記のように、本発明は、一つの物理ネットワーク上に仮想的に複数のネットワークを構成し、各々の仮想ネットワークにそれぞれ異なるセキュリティレベルや通信品質を割り当てることにより、ホストやルータ、スイッチ等で複数のセキュリティレベルや通信品質を利用することが可能となる。

【００２３】また、スイッチ等の機器で接続された機器に対する仮想ネットワークの割当を制限することにより、ホストやネットワークに対して特定のセキュリティレベルや通信品質の利用を制限することが可能である。

【００２４】また、本発明により、“Ethernet VLAN ”や“ATM VC”のような仮想データリンクを利用していないネットワークを流れる通信を上記で実現した様々なセキュリティレベルや通信品質の仮想ネットワークに振り分けることが可能となる。さらに、本発明では、上記で実現したセキュリティレベルや通信品質の仮想ネットワークに応じて、独立したプロトコル処理やアプリケーションの実行が可能になる。これにより、例えば、特定の通信品質の仮想ネットワークのみで稼働するアプリケーションを実現したり、特定のセキュリティレベルではＩＰレベルで中継し、他のセキュリティレベルでは、アプリケーション層で中継したりといった細かい制御を実現したりすることが可能になる。

【００２５】

【発明の実施の形態】まず、以下の説明で用いる、仮想データリンク、仮想ネットワーク、仮想インタフェースの用語を定義する。

【００２６】仮想データリンク（Virtual Datalink: VDL)は、単一の物理ネットワークセグメント内での一つの通信クラスであり、１つのＶＤＬは、それを識別する何らかの識別子が必要になる。識別子の候補としては、次のようなものが考えられる。

【００２７】・８０２．１Ｑ　ＶＬＡＮ tag：この場合、一つのＶＬＡＮｉｄを割り当てられた一セグメントが一つのＶＤＬとなる。

【００２８】・ＩＰのネットワークアドレスとネットマスクの組：この場合、物理インタフェースには複数のＩＰアドレスが付与され、一つのアドレスが一つのＶＤＬを表す。

【００２９】・source port （ソースポート）番号：この場合、source port 番号の特定の範囲が一つのＶＤＬを表す。ＴＣＰのセッションを開く時に、ＶＤＬに応じて適当なsource port を付けたり、ＵＤＰのパケット送出の際に適当なsource port を付ける。仮想ネットワーク(Virtual Network: ＶＮＥＴ）は、複数のＶＤＬを接続して構築される仮想的なネットワークであり、接続する２つのＶＤＬで、その識別子が同じである必要はない。仮想ネットワークは、ルータやゲートウェイを越えて複数のデータリンクに跨がるネットワークの概念である。単一の物理ネットワーク上には複数の仮想ネットワークが存在し得る。

【００３０】仮想インタフェース(Virtual Datalink Interface:ＶＤＩ）は、一つの仮想ネットワーク（ＶＮＥＴ）へのＯＳ内部でのインタフェースである。

【００３１】次に、仮想ネットワークを図面を用いて説明する。

【００３２】図3は、本発明の仮想ネットワークを説明するための図である。

【００３３】同図では、組織内ネットワーク上において、３つの仮想ネットワーク（vnet0,vent1,vnet2)が重なっている。各物理データリンク上では、仮想ネットワーク（ＶＥＴ）が仮想データリンク（ＶＤＬ）として区別されており、区別されたままルータなどで中継されることで複数の物理データリンクにわたる仮想ネットワーク（ＶＮＥＴ）を構成している。

【００３４】このように一つの物理的なネットワークの上に、複数の仮想ネットワークが重なって存在するように見え、多重化された仮想ネットワークの空間を構成す

る。仮想データリンクとして８０２．１Ｑ　ＶＬＡＮを使う場合を考えると、仮想ネットワーク（ＶＮＥＴ）中全域で、ＶＬＡＮtag ｉｄ（ＶＬＡＮタグ識別子）が一意である必要はない。中継するルータの両側で異なるtag ｉｄを用いていても構わない。ここで必要なのは、同一仮想ネットワーク（ＶＮＥＴ）を構成する別物理データリンクの仮想データリンク（ＶＤＬ）間で、tag ｉｄの対応付けが正しく行われることである。つまり、例えば、vnet0 を構成するvdl00 とvdl10について、vdl100 のtag ｉｄが１０で、vdl10 のtag ｉｄが２５であっも構わないということである。

【００３５】同図の仮想ネットワークでは、インターネットへ接続しているルータＧは、vnet0 にしかアクセスできない。ルータＢは、通常のファイアウォールを構成するフィルタリングルータで、インターネットから内部への直接のアクセスを排除するために、vnet0 の外部から通信を遮断している。このファイアウォールセグメント上に外部からのアクセスのために用意されたゲートウェイＡでは、アプリケーションゲートウェイがvent0 からvnet2 への中継を行っている。これにより、以後内部ネットワークにアクセスする際には、vnet2 を通ってアクセスすることになり、例え、遠隔ログインを繰り返しても、内部ネットワークではその通信の起点が外部にあるものと判断できる。

【００３６】次に、ホストやルータ内部でのＯＳ（Operating System）の処理について説明する。

【００３７】図４は、本発明のＯＳ内部の処理を説明するための図である。

【００３８】各仮想ネットワーク（ＶＮＥＴ）に応じてホストやルータ内部での処理を変更するために、ホストやルータ内部のＯＳにおいて、独立な処理モジュールのグループと、それらの複数モジュールグループ間の切替機構が必要になる。ＯＳ内の処理モジュールのグループをResource Space（ＲＳ）と呼ぶ。ＲＳの実体は、処理モジュール中の関数及び、処理モジュール毎のローカルな状態を保持するデータを集めたもので、各々のＲＳは、識別子（ＲＳＩＤ）を持つ。この処理モジュールとは、ＩＰやＴＣＰのプロトコル処理プログラムや、exec()、open()といったシステムコールを受けてＯＳ内部で実際の処理をするシステムコール関数の処理プログラムである。

【００３９】各仮想インタフェース（ＶＤＩ）は、同図中、パケット分別器（packet classifier ）で分別されることにより具現化される（同図中、vent0 やvnet1 の楕円）。ＩＰアドレスやポート番号といった上位層の情報に応じた仮想ネットワークへの振り分けは、分別の対象として上位層の情報を使うことにより、このパケット分別器の分別により実現される。

【００４０】各仮想インタフェース（ＶＤＩ）には、対応するＲＳＩＤが決まっており、受け取ったパケットに対してＲＳＩＤを付属情報として張り付けてＯＳ内部での処理を進める。

【００４１】アプリケーション（同図中一番上の楕円）側から見ると、各プロセスにＲＳＩＤが付いており、システムコールを発呼してＯＳ内部に処理が移った段階で、ＲＳＩＤに応じた処理プログラムが呼び出される。

【００４２】これにより、例えば、“exec()”に対して、“setuid”できない“exec処理ルーチン”を用意しておいて、通信クラスに応じて“setuid”を禁止することなどが可能になる。

【００４３】次に、ＯＳ内部の詳細な構造について説明する。

【００４４】図５は、本発明の多重通信クラス実装時のＯＳ内部の構成を示す図である。同図において、右半分の木構造は、プロトコル処理階層及びシステムコールの概念的な構造を表している。同図中左半分の斜字体の各モジュールは、プロトコル処理モジュール（ｐＩＰ０，ｐＴＣＰ０）やプロコトル依存データ（ｄＩＰ０，ｄＴＣＰ０）、システムコール処理ルーチン（open0，exec0 など）といった処理実体の構造を表している。

【００４５】各ＲＳは、これらのモジュールの集合として実現されるので、ＲＳの実体は、プロトコル処理モジュール／データ、システムコール処理ルーチンへのポインタの集合（同図中、ＲＳn のポインタ表）となる。

【００４６】プロトコル処理モジュールについては、プロトコル依存のデータ（例えば、ＩＰモジュールでは、ＩＰ forwarding するかどうかの変数など）を必要に応じてＲＳ毎に独立にすることにより、同一のプロトコル処理モジュールを使いながら、ＲＳにより処理を変えることも可能である。例えば、同一のＩＰ処理モジュールｐＩＰ0 を用いて、あるＶＮＥＴではＩＰ forwarding を許すが、他のＶＮＥＴでは許さないといったことの実現が可能である。

【００４７】ネットワークから入ってきたパケットは、パケット分別器で分別され、仮想インタフェース（ＶＤＩ：vnet0 等）に渡される。ＶＤＩは、ＲＳに対応付けられているので、ここでＲＳのポインタ表へのポインタがパケットの管理情報として付加されて、以後の処理に渡される。ＲＳが決まれば、処理モジュールの表（ＲＳのポインタ表）が確定するので、以後の処理は、ＲＳの関数とデータを使って進められる。一方、ネットワークにパケットを送出する場合には、プロセスに付加されているＲＳＩＤを使って進められる。一方、ネットワークにパケットを送出する場合には、プロセスに付加されているＲＳＩＤをもとにＲＳのポインタ表を見つけて、ＲＳ毎に処理の切替を実現する。

【００４８】

【実施例】以下、図面と共に本発明の実施例を説明する。

【００４９】ＯＳ内部の処理について説明する。

【００５０】図６は、本発明の一実施例の通信装置の構成を示す。
【００５１】同図に示す装置は、アプリケーション１００、システムコール２００、ＯＳ３００、及びパケット分別器４００から構成される。
【００５２】アプリケーション１００は、プロセス１０１、１０２、１０３、１０４を有する。
【００５３】ＯＳ３００は、ＲＳ３１０，３２０を有し、それぞれのＲＳ３１０，３２０は、プロトコル・処理ルーチン３１１、３２１とシステムコール・処理ルーチン３１２、３２２を有する。なお、同図では、説明の簡単化のため、ＲＳ３１０，３２０の２つのＲＳのみ示してあるが、ｎ個のＲＳが存在する。
【００５４】パケット分別器(packet classifier) ４００は、ネットワークからのパケットの入力を受け、各プロトコル階層のヘッダ情報などを用いて、適切な仮想インタフェース（同図中、vnet0 、vnet1 ）に分別する。分別のための規則は、ルール表４１０の形で当該パケット分別器４００が保持している。このルール表４１０は、ＯＳの構成時に決定され、システムコール２００を通じてアプリケーション１００からの制御で変更が可能である。特定のＲＳ３１０，３２０中のシステムコール処理ルーチン３１２、３２２を、このルール表４１０が変更できないものにしておくことで（同図中のＲＳ３２０のpktctl’ルーチン）、ルール表４１０の制御が不可能なＲＳを作ることが可能であり、このＲＳ３２０上で稼働するアプリケーション１００（プロセス）についてルール表４１０の変更を禁止することができる。
【００５５】パケット分別器４００で仮想インタフェースに分別されたパケット（vnet0,vent1)は、その仮想インタフェースに対応するＲＳ３１０，３２０のプロトコル処理ルーチン（ＴＣＰ，ＵＤＥＰ，ＩＰ，ｒｔ等）３１１、３２１に渡される。プロトコル処理ルーチン３１１、３２１は、ＩＰ，ＴＣＰ，ＵＤＰ等のプロトコルの実際の処理をするルーチンである。このプロトコル処理ルーチン３１１、３２１は、ＲＳ３１０、３２０毎に保持するポインタ表３１３、３２３で管理されており、プロトコル処理ルーチン間の関係もこの表で決定する。
【００５６】アプリケーション１００からの処理要求は、システムコール２００を介して処理される。システムコール２００の引数として、ＲＳＩＤが渡される場合は、システムコール処理ルーチン（ｏｐｅｎ，ｅｘｅｃ等）３１２、３２２が呼び出される前に、そのプロセスが指定されたＲＳの処理を使ってよいかどうかの許可表３３０を検索し、権限を調べる。ＲＳＩＤが渡されない場合は、プロセスに付与されているＲＳＩＤを用いる。システムコール処理ルーチン３１２、３２２は、ＲＳＩＤをもってＲＳ対応表３４０検索し、該当するＲＳのポインタ表３１３、３２３を獲得することで、当該ポインタ表３１３、３２３に指し示されているシステムコール処理ルーチン（ｏｐｅｎ，ｅｘｅｃ等）３１２、３２２を呼び出す。
【００５７】以下に、上記の構成における処理を説明する。
【００５８】アプリケーション１００からネットワークやディスク等の周辺機器に対する読み書きの処理は、システムコール２００を通じて行われる。システムコールは、アプリケーションとＯＳ内部の処理の切り替えを行うインタフェースである。
【００５９】図７は、本発明の一実施例のシステムコールの処理を示すフローチャートである。同図に示す処理は、ソケットやファイルのオープン、読出し要求、書込み要求など全てのシステムコールの処理に共通する。
【００６０】まず、アプリケーション１００がシステムコール２００を呼び出す（ステップ１０１）。次に、ＲＳＩＤが引数に指定されているかを確認し（ステップ１０２）、指定されていない場合には、アプリケーション１００のプロセスに付与されているＲＳＩＤを調べ（ステップ１０３）、ＲＳＩＤから該当のＲＳのポインタ表（３１３、３２３）を特定し、システムコール処理ルーチン（３１２、３２２）を呼び出す（ステップ１０５）。指定されている場合には、許可表３３０でそのプロセスに許可されているＲＳかを判定し（ステップ１０４）、ＲＳＩＤから該当のＲＳのポインタ表（３１３、３２３）を特定し、システムコール処理ルーチン（３１２、３２２）を呼び出す（ステップ１０５）。
【００６１】次に、ネットワークから入力されたパケットの処理について説明する。
【００６２】図８は、本発明の一実施例のネットワークからの入力処理のフローチャートである。
【００６３】ネットワークからパケット分別器４００にパケットが到着すると（ステップ２０１）、データリンク層、ネットワーク層、トランスポート層などのヘッダを見て、ルール表４１０を検索する（ステップ２０２）。検索した結果該当するものがあれば（ステップ２０３，Ｙｅｓ））、該当したルールに対する仮想インタフェースを用い（ステップ２０４）、該当するものがなければ（ステップ２０３、Ｎｏ）、デフォルトの仮想インタフェースを用いる（ステップ２０５）。次に、仮想インターフェースを通じてＲＳのポインタ表（３１３、３２３）を設定し、プロトコル処理ルーチン（３１１、３２１）にパケットデータを渡す（ステップ２０６）。プロトコル処理ルーチン（３１１、３２１）は、ＲＳのプロトコルスタックに応じてパケットを処理し（ステップ２０７）、アプリケーション１００の読出し（システムコール）を待つ（ステップ２０８）。
【００６４】上記の流れで処理されたデータは、読出しのシステムコール２００を通じてアプリケーション１００に渡される。
【００６５】次に、アプリケーション１００からネット

ワークに書き出されたデータの処理について説明する。

【００６６】図９は、本発明の一実施例のネットワークへの出力処理のフローチャートである。

【００６７】システムコール２００などを通じてデータが書き込まれ（ステップ３０１）、システムコール処理ルーチン（３１２、３２２）の属するＲＳ３１０、３２０から、プロトコル処理ルーチン（３１１、３２１）が特定されると（ステップ３０２）、プロトコル処理ルーチン（３１１、３２１）でデータを処理し（ステップ３０３）、ＲＳ３１０、３２０から特定される仮想インタフェースを通じてデータ（パケット）をネットワークに送出する（ステップ３０４）。

【００６８】上記の処理において、データはシステムコール２００を通じてＯＳ３００内の処理ルーチンに渡され、プロトコル処理を通ってネットワークに送出される。

【００６９】上述のように、本発明では、ＯＳやサービスレイヤで、通信トポロジを管理すると共に、仮想データリンクを個別認識し、独立に中継することにより、通信の起点に応じてアクセス制御を行うことができ、ネットワークのセキュリティを容易に高めることが可能となる。

【００７０】なお、本発明は、上記の実施例に限定されることなく、特許請求の範囲内において種々変更・応用が可能である。

【００７１】

【発明の効果】上述のように、本発明によれば、一つの物理ネットワーク上にセキュリティレベルや通信品質の異なる複数の仮想的なネットワークを構成し、それらの仮想ネットワーク毎に適切なプロトコル処理やアプリケーション実行が可能になる。これにより、例えば、インターネットに晒された比較的危険なネットワークや、社内の多数のユーザがアクセスするネットワーク、部署内のユーザだけがアクセスするネットワーク、一人のユーザだけがアクセスするネットワークといった複数のセキュリティレベルに応じた仮想ネットワークを用意し、それぞれに稼働させるアプリケーションを制限したり、通信の中継方法をＩＰ層の中継やアプリケーション層の中継など多様に使い分けることにより、ネットワークに対するセキュリティをきめ細かに制御することが可能になる。

【００７２】また、これらの仮想ネットワークの提供をスイッチ等のネットワーク機器側から制御することも可能となる。これにより、初心者から上級者に至るまでの様々な技術のユーザに応じた、適切なインターネットアクセス及びイントラネットアクセスを提供することができる。

【図面の簡単な説明】

【図１】本発明の原理を説明するための図である。

【図２】本発明の原理構成図である。

【図３】本発明の仮想ネットワークを説明するための図である。

【図４】本発明のＯＳ内部の処理を説明するための図である。

【図５】本発明の多重通信実装時のＯＳ内部の構成図である。

【図６】本発明の一実施例の通信装置の構成図である。

【図７】本発明の一実施例のシステムコールの処理を示すフローチャートである。

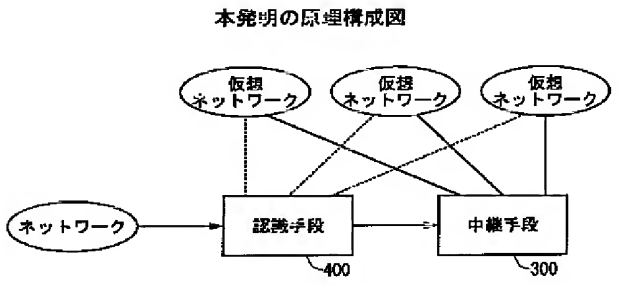
【図８】本発明の一実施例のネットワークからの入力処理のフローチャートである。

【図９】本発明の一実施例のネットワークへの出力処理のフローチャートである。

【符号の説明】

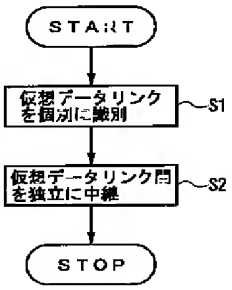
１００　アプリケーション
１０１、１０２、１０３、１０４　プロセス
２００　システムコール
３００　ＯＳ，中継手段
３１０，３２０　ＲＳ（Resource Space）
３１１，３２１　プロトコル処理ルーチン
３１２，３２２　システムコール処理ルーチン
３３０　許可表
３４０　ＲＳ対応表
４００　認識手段、パケット分別器
４１０　ルール表

【図２】



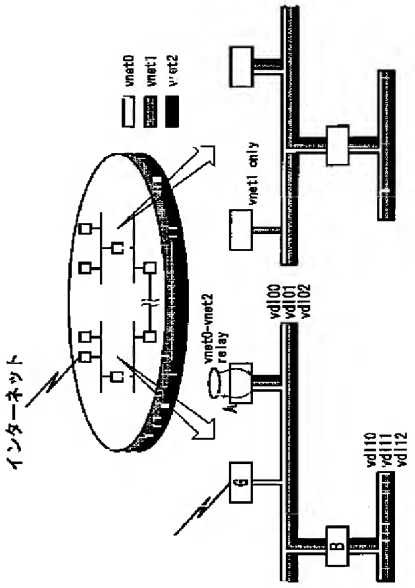
【図1】

本発明の原理を説明するための図



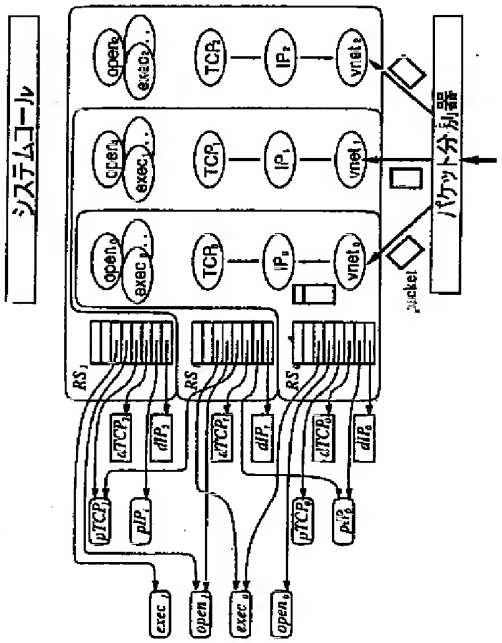
【図3】

本発明の仮想ネットワークを説明するための図



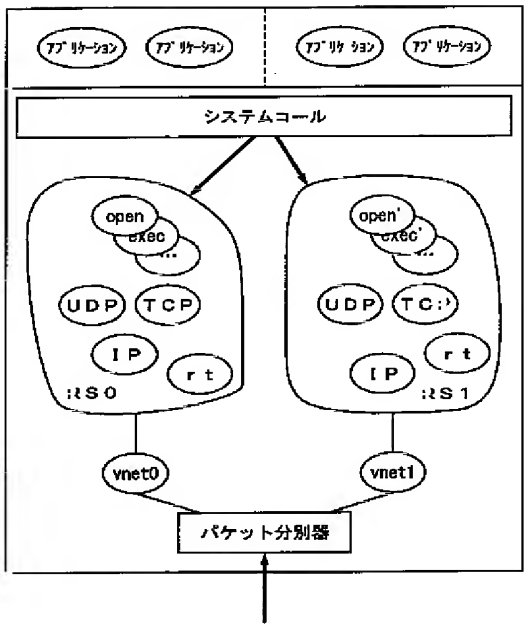
【図5】

本発明の多重通信実装時のＯＳ内部の構成図



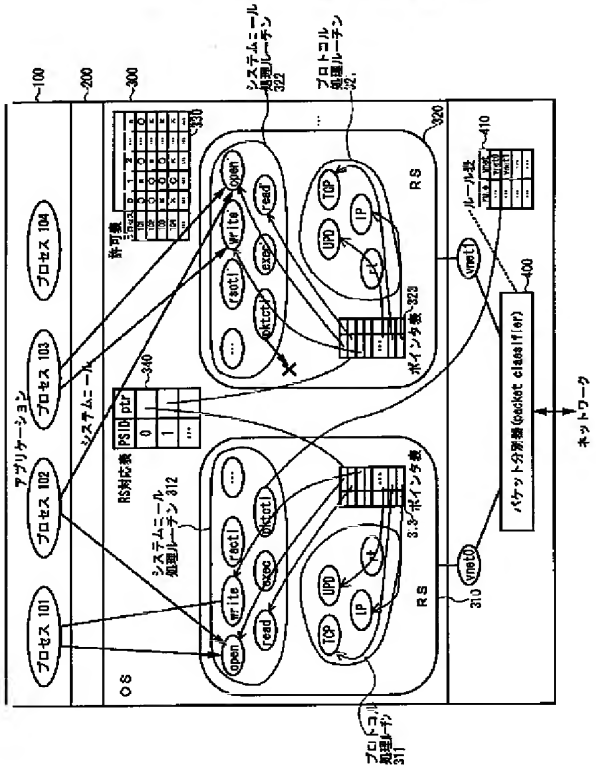
【図4】

本発明のＯＳ内部の処理を説明するための図



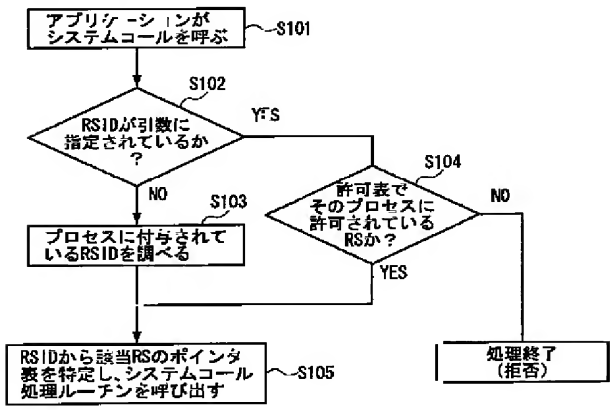
【図6】

本発明の一実施例の通信装置の構成図



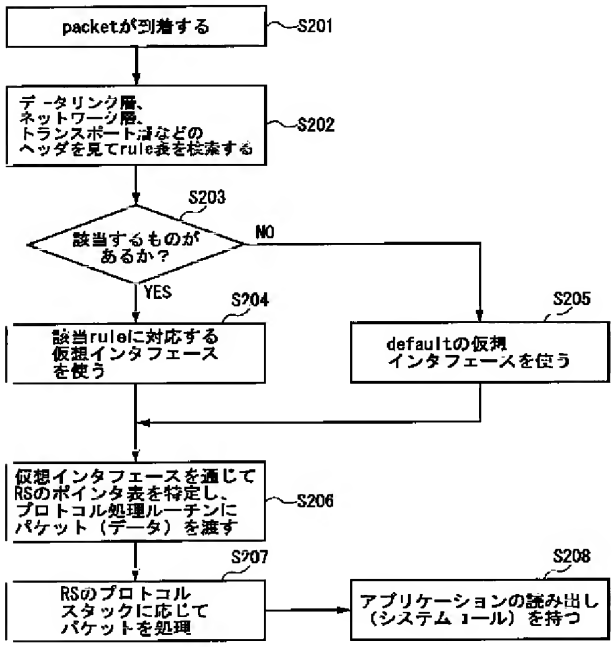
【図７】

本発明の一実施例のシステムコールの処理を示すフローチャート



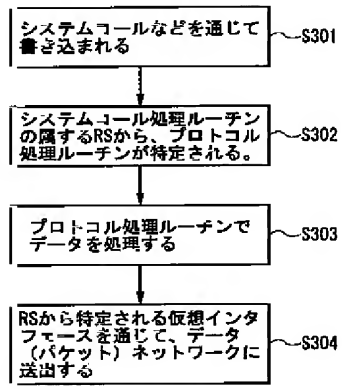
【図８】

本発明の一実施例のネットワークからの入力処理のフローチャート



【図９】

本発明の一実施例のネットワークへの出力処理のフローチャート



フロントページの続き

(51)Int.Cl.[7]	識別記号	ＦＩ	(参考)
	Ｈ０４Ｌ　29/04		

(72)発明者　明石　修
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　佐藤　孝治
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　山崎　憲一
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　菅原　俊治
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内

Ｆターム(参考)　5B089 GA04 GB01 HB19 KA04 KB03
KB06 KF06 KG05 KG08
5B098 GA02 GA04 GC01
5K030 HC01 HC13 HD03 LB05
5K033 BA04 CB08 CC01 DA05 DB18
5K034 JJ24 KK21 KK27 KK28